Available online at www.sciencedirect.com

SciVerse ScienceDirect

Procedia - Social and Behavioral Sciences 64 (2012) 209 – 218

Procedia
Social and Behavioral Sciences

IETC 2012

International Educational Technology Conference, Taoyuan, Taiwan, 11-13 Jul. 2012

A Time and Cost Effective Approach to 3D Scenario Building in a Digital Game Design Class

Ming-Shiou Kuo[a,*] , Tsung-Yen Chuang[a]

[a]Department of Information and Learning Technology, National University of Tainan, Tainan County,700, Taiwan

Abstract

This paper demonstrates an integration of modern technologies to provide an effective approach that can help teachers and students focus on creative design rather than on complex graphic tools or commercial game engines in a capstone class. Ignoring the unnecessary functions of expensive commercial packages, we develop and implement our own solution package that includes Google SketchUp, 3D Warehouse and Microsoft Virtual Worlds technology to teach gaming context and content design with 3D scenario building. The empirical experiment proved the feasibility of our work when the students were able to visualize and present their designs of digital games with less cost and fewer efforts than would be needed with the commercial software packages that are generally used to achieve this.

Keywords: Digital Game Design, 3D Scenario, Google SketchUp, 3D warehouse

1. Introduction

There has been an increasing interest in game-based learning and bringing digital game design into the classroom due to the widespread popularity of such games, and particularly of 3D multi-user role-playing games, which are based online and allow the players' avatars to exist in realistic virtual environments.

* Corresponding author. Tel.: +886-8-7703202; fax: +886-8-7740165
E-mail address: ccmitchellkuo@gmail.com

doi:10.1016/j.sbspro.2012.11.025

However when teachers try to help students who are interested in designing 3D gaming worlds, or aim to enter the digital gaming industry after graduation, they face a number of challenges. These include the budget that is available for such classes, which influences the decision about which software to use for training, and the time available for both delivering design theories as well as assessing students' performance.

From the students' perspective, the first challenge they may encounter when attempting to create a 3D game design is the significant efforts that are required to learn how to use the 3D modeling software to create game characters, buildings and objects, such as MAYA or 3DS Max from Autodesk™, as well as the high cost of such packages if they wish to use them outside of class. Although there are arguments about the value of teaching 3D modeling in a design class, exposure to it is essential preparation for many careers, and it can also expands students' design-related thinking (Cockburn & McKenzie, 2001). Research on student perceptions of game-based learning courseware found that the majority (58.2%) favored 3D animation over 2D (22.6%) (Zin, Yue, & Jaafar, 2009). Although in Karl's research the skills required for carrying out games design (Jeffries, 2011) using 3D Studio Max were not seen as important by any subject of the academic respondents, the ability to present both the game and players in a public context, so that they can be assessed, was seen as very important by the respondents who were practitioners. Our argument in the current work is thus that visualizing the design of the game in some form that can undergo peer reviews will improve the quality of a design class, if the 3D learning curve is well controlled.

Teachers of digital game design classes thus need to overcome the following challenges: preventing the teaching of complex 3D software issues from taking up limited credit hours in a class that should spend more time on game design; obtaining the budget needed to teach effective classes; and dealing with the frustration that students feel if they do not understand why they have to spend a lot of time and effort carrying out 3D modeling instead of focusing on the theoretical and creative aspects of game design.

Furthermore, even the students have successfully prepared their collections of 3D objects or digital content, they need a virtual stage to practice and eventually demonstrate their ideas about game stories and scenarios, as well as prototypes, so that both the teacher and classmates can review and comment on them. They thus face problems with regard to the expense of buying commercial game engines, such as Unreal™, Virtools™, and Torque Engine™, in addition to the learning efforts required, which will make heavy demands on their time. In Albert's study on creating a game development course with limited resources (Ritzhaupt, 2009), which used the Torque Engine™, indicated that the cost and difficulty of the selected platform was seen as 5 on a 5-point scale, even higher than the 4 points for the difficulty in learning how to use the 3D software, again highlighting the problems of high costs and steep learning curves.

These concerns were also raised by students in another study (E. Vincent Cross et al., 2008). All the students surveyed in the article noted that the expense and difficulty of the game engine was the biggest problem that they faced on the course. One respondent even stated that while their participation in the course was empowered by their creativity, it was hindered by their limited resources. A number of subjects in the study suggested that the course should be split into two separate courses, the first part being theory and the second part practice. However, the strategy of trading time for resources is sometimes limited by teachers, the policies of the related institutions, or other factors.

This work aims to help teachers and students focus on their creative design activity rather than software operations, and to reduce the need to buy expensive software when only a few of its functions are used in the class.

2. Design and development

2.1. The groundwork

This study is based on two elements: enhancing student creativity with regard to 3D game design, and developing a platform for students to practice and demonstrate what they have learned in class. The major considerations are the expense and the ease of use of the software.

2.2. 3D modeling software

3DS Max and Maya are the most commonly used 3D modeling, animation and rendering software, they are often very difficult to learn how to use, especially for teachers or students who do not have a digital arts or computer science background. However, Google's free SketchUp 3D modeling software is relatively intuitive to use, and thus has attracted more and more users from various disciplines. Moreover, the accompanying free 3D warehouse website (Google, 2012) provides a huge collections of 3D models, which means that students do no need to build their own content from the very beginning, but can instead modify existing designs, as shown in Fig. 1, and this can save a great deal of time and money.

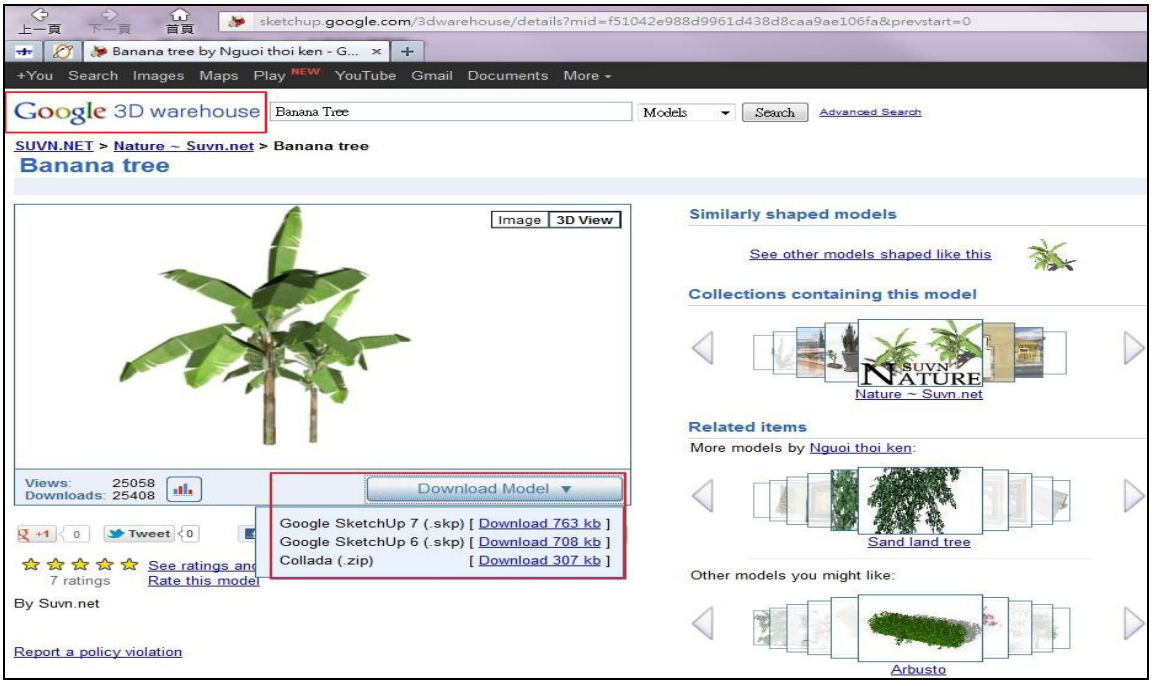

Fig. 1. The Google 3D warehouse provides many free and downloadable 3D models that students can then modify

So that it can work with our presentation platform, which will be discussed in the next section, the output from SketchUp is converted into the Microsoft DirectX format by the free 3D RAD exporter plugin (3DRAD.com, 2012). Most students in Taiwan use the Windows™ operating system, and thus the DirectX file format is a viable one. Moreover, when in text mode it can be easily edited by text editing applications like Notepad, which is bundled with Windows. Using this system, students can easily replace the texture files of a 3D object and quickly see the new effects, and then save the object into a binary compressed format to reduce the file size for easier online transmission. This process is shown in Fig. 2.

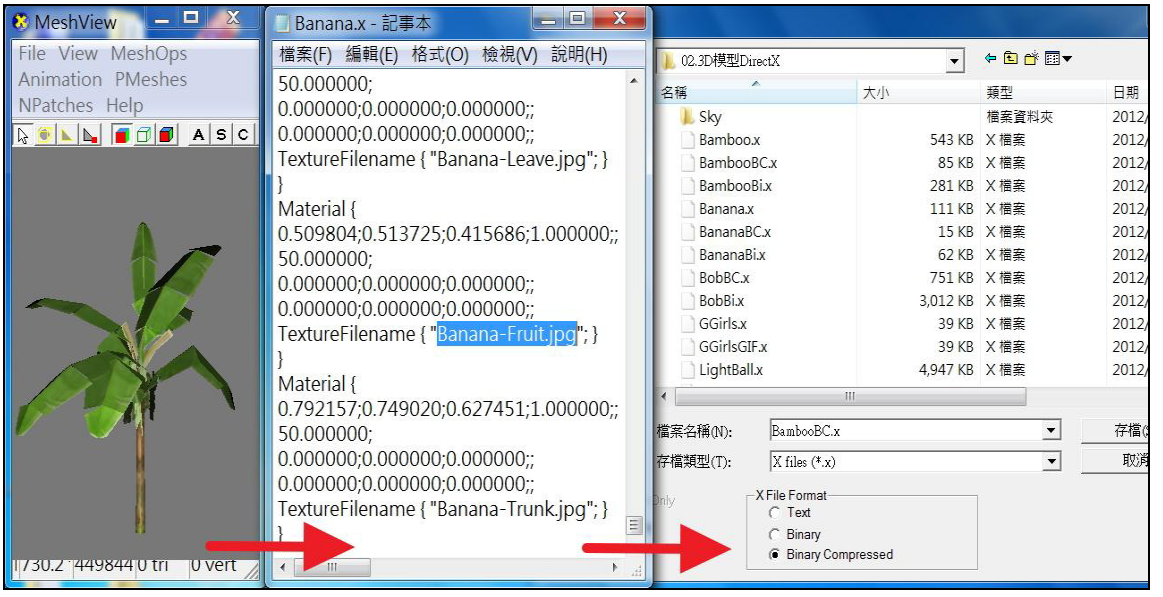

Fig. 2. Using a text editor to replace the texture file of a 3D object, and then compress the file size using the free DirectX Mesh Viewer

In addition, we develop an http handler for the 3D objects so that students can retrieve their texture files from any remote web server via the http protocol that uses TCP/IP port 80. Part of the code for this is shown below:

..... MeshMaterialList{1; 2; 0, 0;;Material {1.000000;1.000000;1.000000;1.000000;;3.200000;0.000000;0.000000;0.000000;;0.000000;0.000000;0.000000;;TextureFilename {
"http://RemoteServerIP/getImage.ashx?ImageUrl=/Upload/StudentNo/TextureFor3DObject01.jpg";}
……

This can help game designers change the appearance of distributed 3D objects easily by replacing the texture files which are stored on one or more central websites, without repacking and redistributing the whole files, which may be very large and consume too much bandwidth.

2.3. The platform solution

Before recommending a cheaper and easier to use platform for students, we tested some commercial engines, such as the Virtools, Unity3D, Second Life, and so on. In addition to reducing the price and

learning curve, the following characteristics are suggested for any platform: multi-user capability for remote and collaborative peer reviews; the ability to meet firewall criteria in most educational networks, which block various gaming ports to avoid bandwidth abuse; as few steps as possible needed for the students to import and manipulate their 3D objects; extensibility to ensure that the system can evolve to meet future needs.

After examining the currently available software and developing the criteria listed above, we chose to develop our own solution based on the Microsoft Virtual Worlds research project (Vellon, Marple, Mitchell, & Drucker, 1998). Although the source code for this software is out of date, it is free of charge for teaching purposes and compatible with the Windows operating system. Moreover, we modified the source code to meet the aims of this study. First of all, the kernel for the supporting operating system was modified from Windows 95 to the more recent Windows 7, and there was also a move from IE 5.5 to IE 9 for the user interface, and from Windows 2000 to Windows 2008 for the 3D multi-user server. Second, in addition to using the http port 80 to distribute 3D and other web contents over the Internet, we modified and compiled the source code of the 3D multi-user server so that it could communicate with multiple clients via port 443, which is very widely used online. These modifications meant that this software was able to comply with most of the firewall rules used on campus networks, thus allowing a collaborative virtual environment. Last but not least, the client user interface was redesigned using a browser and HTML, as this means the text of the menu items and tool tips can be easily translated into other languages, such as Traditional Chinese for our subject students, as shown in Figs. 3 and 4.

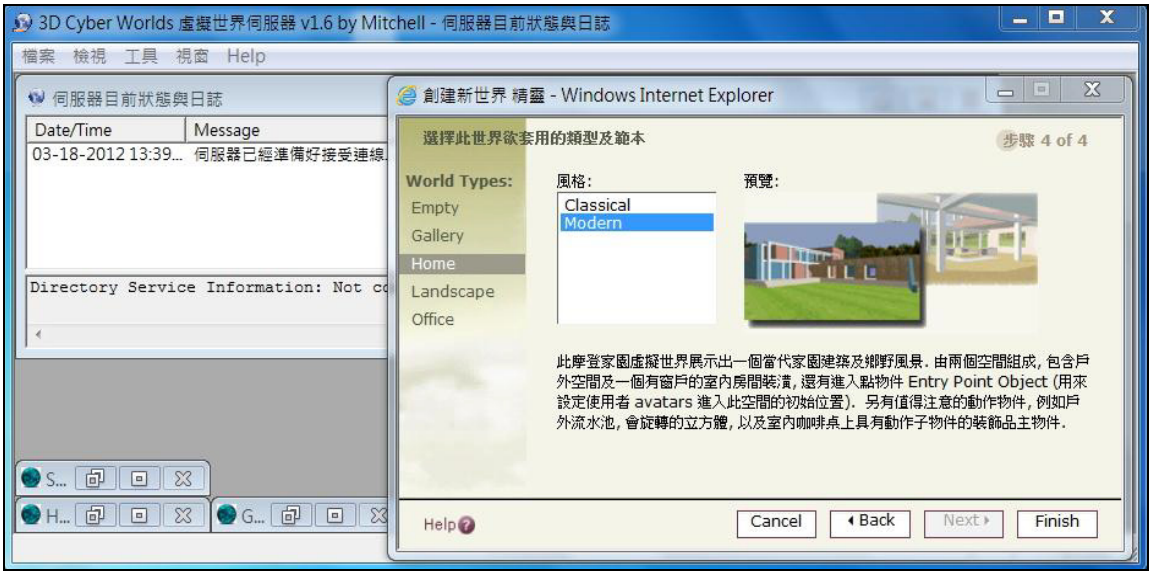

Fig. 3. Using the IE browser as the client interface, so that the text can easily be translated by editing the HTML

Fig. 4. The user interface to organize 3D content

Because ActiveX, which was used as the API, and the IE browser, which was used as the interface and bundled with Windows, this approach has good extensibility and can work with many other online applications. There are more programmers capable of using ASP.Net, HTML5 and other mainstream web technologies than there are who can use certain commercial platforms, such as Virtools or Unreal 3D. In addition, commercial game engines sometimes focus on only one genre, such as first-person shooter games. Figure 5 shows an online multi-learner activity applying our solution, which integrates ASP.Net, Ajax, Adobe Flash and YouTube video, and thus our approach is easy to operate for people with general web programming skills, and can easily be expanded with other browser-based technologies, such as internet-based voice conferencing. Besides, the feasibility of this system has been demonstrated in educational contexts, such as game-based learning projects (Kuo, Chuang, Lin, & Chou, 2011; Kuo & Lin, 2010; Kuo, Lin, Ma, & Chen, 2009) and it has also shown potential for being used in digital game design classes.

Fig. 5. The platform integrates ASP.Net, AJAX, Flash and other web-based technologies

The basic physical deployment of our system is shown in Fig. 6. The clients use web browsers and a web portal to connect to the virtual 3D environments which are generated by one or several distributed 3D multi-user platforms, as introduced above. The database server can store all of the information that is needed to carry out any of the associated activities.

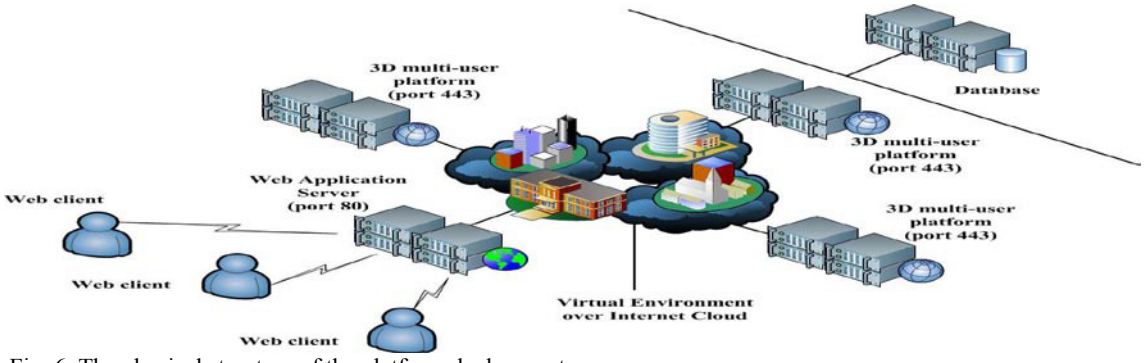

Fig. 6. The physical structure of the platform deployment

The practical implementation of the system can be arranged as in Fig. 7, which shows a 3D collaborative online learning activity. Supported by forums, avatars, instant messages and various other applications, learners are able to gather in a virtual learning space to carry out their learning tasks.

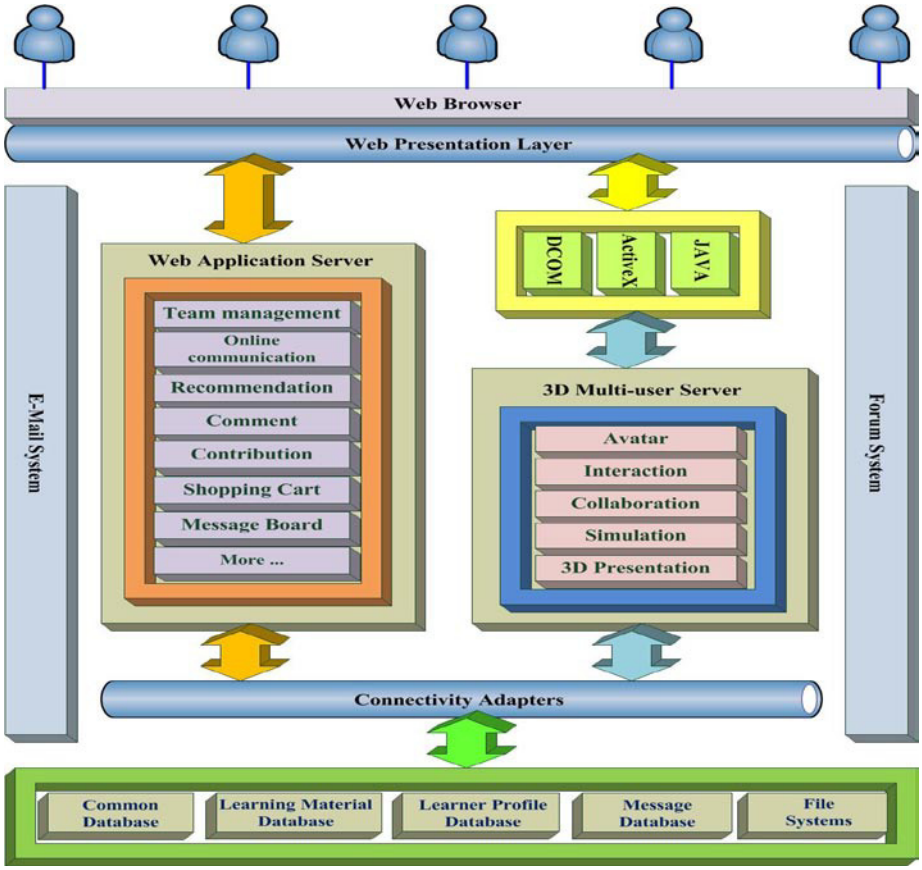

Fig. 7. An example of an implementation of the system

An experiment for testing system feasibility was run first at the campus of the National Pingtung University of Science and Technology (NPUST), with seven juniors from the Department of Management Information Systems. Subject students were told to begin their design work after watching a pre-recorded series of video tutorials which were hosted on the E-learning website of NPUST and called “The practice of 3D internet multi-user environments.” The total length of the video series is just under five hours, including about 45 minutes introducing the basic principles of using Google SketchUp, and two hours of assigned hands-on labs activities.

2.4. The results and implementation

After the online video instruction and the system feasibility were tested at NPUST, the system is now being adopted in a digital game design class with 40 sophomores and three seniors from the Department of Information and Learning Technology, three juniors from the Department of Fine Arts, two juniors and one senior from the Department of Computer Science and Information Engineering, and one senior

student from the Department of Drama Creation and Application at the National University of Tainan (NUTN). The students are divided into 12 groups of four or five members each for brain storming and collaborative game design. Each week, starting in the ninth week, two groups are required to demonstrate their designs to the rest of the class. The theoretical part of the instruction is delivered in lectures by the teacher in the class, while the practical part is introduced only in the first week and then supported by the online learning materials, which students can repeatedly view via network or by offline DVD videos. In addition, students can post their questions and comments on a course discussion board.

3. Conclusions

The purpose of this study was to find a more efficient method for teachers and students to work on projects in a 3D game design class. The system developed in this work can help the feasibility of students' game designs be discussed in more detail before investments in valuable class time or expensive and complicated software. For 3D manipulation, the system uses DirectX in text format so that the students can easily change the texture of a 3D object before compressing the object into a smaller file size in binary format in order to share the results. By taking advantage of Google's free yet powerful 3D modeling tool, along with its free 3D warehouse website, students can save both time and money when building 3D models.

For the digital presentation stage, our system uses the free Microsoft Virtual Worlds technology, as this can reduce the costs and learning curve associated with a 3D digital game design class. The extensibility and maintenance of our system are also considered, and these are not problematic due to the use of popular web- and internet-based programming languages, rather than dedicated, commercial game engines. In addition, the TCP/IP port the system used was set at 80 and 443, as these work well with academic networks, which often block the gaming ports used by commercial game engines.

The prototype of our solution was successfully tested at NPUST, and a full scale implementation is now being carried out in a digital game design class at NUTN. The 3D design software and presentation platform are both free and easy to use, and teachers and students can review and comment on every design produced by the class without needing to make large investments of time, money or efforts.

References

3DRAD.com. (2012). Exporting from Google SketchUp to Directx/XNA (.x) file format. Retrieved March 21, 2012, from http://www.3drad.com/Google-SketchUp-To-DirectX-XNA-Exporter-Plug-in.htm

Cockburn, A., & McKenzie, B. (2001). *3D or not 3D?: evaluating the effect of the third dimension in a document management system*. Paper presented at the Proceedings of the SIGCHI conference on Human factors in computing systems, Seattle, Washington, United States.

E. Vincent Cross, I., Gosha, K., Eugene, W., Arcediano, F., Hamilton, C., & Hundley, J. (2008). *Game design from the lens of a student*. Paper presented at the Proceedings of the 46th Annual Southeast Regional Conference, Auburn, Alabama.

Google. (2012). Google 3D warehouse Retrieved March, 2012, from http://sketchup.google.com/3dwarehouse/

Jeffries, K. K. (2011). Skills for creativity in games design. *Design Studies, 32*(1), 60-85. doi: 10.1016/j.destud.2010.07.001

Kuo, M.-S., Chuang, T.-Y., Lin, C.-S., & Chou, C.-C. (2011, November 28, 2011 (Monday) to December 2, 2011 (Friday)). *From Speaking to Acting - Enhancing Interactivities of Learning in the 3D Virtual Worlds.* Paper presented at the The 19th International Conference on Computers in Education, ICCE 2011, Chiang Mai, Thailand.

Kuo, M.-S., & Lin, C.-S. (2010). *Virtual Parabola Festival: The Platform Design and Learning Strategies for Virtual Learning Community of Practice*. Paper presented at the Digital Game and Intelligent Toy Enhanced Learning (DIGITEL), 2010 Third IEEE International Conference Kaohsiung, Taiwan.

Kuo, M.-S., Lin, C.-S., Ma, J.-T., & Chen, Y.-L. (2009). *The Impact of Learner Profile and Collaborative Knowledge Construction on Virtual Learning Community of Practice.* Paper presented at the Global Chinese Conference on Computers in Education, GCCCE 2009, Taipei, Taiwan.

Ritzhaupt, A. D. (2009). Creating a Game Development Course with Limited Resources: An Evaluation Study. *ACM Transactions on Computing Education, 9*(1), 1-16. doi: 10.1145/1513593.1513596

Vellon, M., Marple, K., Mitchell, D., & Drucker, S. (1998, April 27-30). *The Architecture of a Distributed Virtual Worlds System.* Paper presented at the The 4th USENIX Conference on Object-Oriented Technologies and Systems (COOTS), Santa Fe, New Mexico.

Zin, N. A. M., Yue, W. S., & Jaafar, A. (2009). Digital game-based learning (DGBL) model and development methodology for teaching history. *WSEAS Transactions on Computers, 8*(2), 322-333.